Collaborator: W. Cazemier (NRA) 
Ellingson,죠산.edu
1320 kinmear Rd, Columbus, OH 43212
The Ohio State University Electrosence Laboratory

Steve Ellingson

February 29, 2000
Arecibo Observatory, PR

for SKA-class systems
Joint Multibeaming and RFI Suppression
• The resulting joint FFT/multi-forming implementation has a computational cost than is only a little higher than FFT alone.

• This null-forming technique has been extended for use in FFT-based multi-forming.

• But currently designed for single beams, a useful large-array null-forming technique is available.

• For RFI suppression, FFT beaming is desirable for lower computational cost.

• RFI suppression

• Wide field-of-view

• Various "SKA-class" systems now being planned.

Introduction
Beam pointed to broadside (90°)

Uniform linear array of \( N = 16 \) elements, \( \lambda/2 \) spacing

Conventional Beamforming (CBF)
as the CBF

Once $y$ is known, this modified beamformer has the same computational cost

$$a_H(t^T d) = w$$ where $w = (t)(t)\hat{y}$

Beamformer can be redefined as

$$\hat{y} = \text{nulling recipe out the interference subspace estimate}$, that’s nulling!

$$p_{H_{\hat{y}} \hat{y} - (s_{H_{\hat{y}}}^s \hat{y}) \hat{y} - 1 = \hat{y}^T$$

$\hat{y}_{\hat{y}} \hat{y}$

1. $\hat{y}$

Endcomposition

(covariance matrix)

$\{(t)_{H}^x(t)_{x}\} \mathbb{F} \rightarrow \hat{y}$

One way to calculate $\hat{y}$ is as follows (Subspace Projections):

$x(t)$ is the vector of antenna outputs

$x(t)$ is the vector of weights for desired beam (Steering vector)

$\hat{y}$ is the vector of weights for desired beam (Steering vector)

$$\hat{y} = (t)\hat{y}$$

Nulling by Spatial Projections
- Solid: Modified CBF, Dash: CBF
- Negligible distortion to pattern away from null
- Single null placed at 20°
\{(t)x\} \xrightarrow{\text{FFT}} (f) \quad (f) = (f) x \quad (f)

Then, one can invoke the Fast Fourier Transform (FFT):

\text{MATLAB: \texttt{fft}(N)}

where \( P \) is the matrix--operator of the Discrete Fourier Transform (DFT).

If the antennas fall along a rectilinear grid, one choice for \( P \) is

\[ a^w \quad \text{are the desired steering vectors} \]

\[ N \dot{\geq} \quad \mathcal{W} \quad \text{is the beamforming matrix; e.g.,} \quad \mathcal{W} = \begin{bmatrix} \mathcal{W} \ldots \mathcal{W} \end{bmatrix} \]

\( \text{B is the vector of beamformer outputs} \)

\[ y(t) = B x(t) \]

General MultiBeamforming: \( y(t) = B x(t) \)

MultiBeamforming
(Only the center-most 7 of 16 beams shown)

\[
\text{FLOPS} \propto N \log_2 N
\]

*FFT*

*DFT*

**DFT and FFT Multibeamforming**
But, there is no fast transform version of the joint processor in this case.

Therefore, multibeamforming alone, in the general matrix beamformer case, multibeamforming + multilink has almost the same computational cost as

\[ \mathbf{G} = \mathbf{B} \mathbf{P} \]

Faster: \( \mathbf{y}(t) = \mathbf{G} \mathbf{x}(t) \), where

General multibeamforming with subspace projection: \( \mathbf{y}(t) = \mathbf{B} \mathbf{P} \mathbf{x}(t) \)

General multibeamforming: \( \mathbf{y}(t) = \mathbf{B} \mathbf{x}(t) \)

Multibeamforming with spatial projections (general case)
Joint DFT Multibeaming + Spatial Projection

- Single null placed at 20°
- Note null is well-formed for all beams
- Negligible distortion to patterns away from null
this additional cost is small compared to the cost of $\text{FFT}^{-1}$ then the interference is slowly varying (so the cost of updating $\text{FFT}^{-1}$ is small),

If $\lambda \gg N$ (number of desired nulls is much less than number of elements), and so at the cost of $\lambda$ additional beamformers, one can have joint $\text{FFT}$ and nullforming

\[
\begin{align*}
\{\mathbf{1}\} \mathsf{FFT} & \begin{bmatrix} \mathbf{1} \times 2 \mathbf{H} ^{\top} \mathbf{n} \end{bmatrix}_{} ^{=\top} - \{\mathbf{1}\} \mathsf{FFT} = (\mathbf{1}) \mathbf{\Lambda} \\
\{n\} \mathsf{FFT} & \begin{bmatrix} \mathbf{1} \times 2 \mathbf{H} ^{\top} \mathbf{n} \end{bmatrix}_{} ^{=\top} - \{\mathbf{1}\} \mathsf{FFT} = (\mathbf{1}) \mathbf{\Lambda} \\
\mathbf{H} ^{\top} \mathbf{n} \mathbf{n}^{=\top} _{} ^{=\top} \mathbf{I} - \mathbf{I} = \tau \mathbf{d} \\
\mathbf{H} ^{\top} \mathbf{n} \mathbf{n}^{=\top} _{} ^{=\top} = \mathbf{d}
\end{align*}
\]

Can this be made to run with $\text{FFT}$-like computational cost? Consider:

\[
\{\mathbf{1}\} \mathsf{FFT} \mathsf{Multibeamforming} \leq \text{Subspace projection: } \mathbf{\Lambda} \mathbf{\Lambda} = \mathbf{\Lambda} (\mathbf{1}) \mathbf{\Lambda}
\]

**FFT Multibeamforming with Spatial Projections**
(here, $\phi = 1$)

\[ \text{FLOPS} \propto N \log_2 N + \frac{\mu N}{N} \] (approx.)

Modified FFT

Joint FFT Multibeamforming + Spatial Projections
Low pattern distortion away from nulls (deterministic, too)

- Nulls placed at 20°, 30°, 45° (γ = 4)

Same "modified FFT" developed in previous slides

Example with 4 Nulls Placed
Closing Remarks